UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/663,866	09/15/2003	Deepak Ayyagari	8371-156	3126

46404 7590 07/30/2010

MARGER JOHNSON & MCCOLLOM, P.C. - Sharp
210 SW MORRISON STREET, SUITE 400
PORTLAND, OR 97204

EXAMINER	
WU, JIANYE	

ART UNIT	PAPER NUMBER
2462	

NOTIFICATION DATE	DELIVERY MODE
07/30/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docketing@techlaw.com

Office Action Summary	Application No.	Applicant(s)	
	10/663,866	AYYAGARI, DEEPAK	
	Examiner	Art Unit	
	JIANYE WU	2462	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-9,11,13 and 21-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-9,11,13 and 21-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendments/Arguments

1. Applicant's arguments filed on 3/31/2010 have been fully considered, Some of arguments are persuasive. However, upon further consideration, new ground rejections are made.

2. For claims 17-19, Applicant makes argument "classifier rules were previously introduced in dependent claims 17- 19, the rejection of claim 1 will be addressed in light of the rejections of claims 17-19 as well" (2nd paragraph of page 8).

In response, Examiner would like to point out that claims 17-19 have been cancelled.

For **claim 1**, Applicant argues: "The Examiner apparently identified a 5-tuple in Stevens as the classifier rule. The 5-tuple identifies a protocol, a local address, a local process, a foreign address and a foreign process. See Stevens, p. 269, 1. 8. However, these parameters do not include matching criteria and a connection identifier as described above, nor is this 5-tuple applied to application data" (3rd paragraph of page 8);

In response, Examiner would like to point out that Office Action does not use 5-tuple to address the classifier rule in claim 1. Claim 1 has been significantly amended, and they are addressed in the office action. Please see the office action for more details.

3. For **claim 6**, Applicant argues:

a) "The Examiner noted that Tanenbaum did not teach the determination of the order of the rules. See Office Action, p. 10. The Examiner cited Figure 6.7 of Stevens to teach the rules. See Office Action, p. 11. However, Figure 6.7 is only a list associating various combinations of family, type, and

protocol with an actual protocol. Nowhere is there a reference to a priority. The Examiner is apparently arguing that data transmitted through a stream socket has a higher priority than data transmitted through a datagram socket; however, there is no such teaching in Stevens. Thus, the Examiner is apparently relying on his own knowledge to teach such a feature." (paragraph 3-4, page 10);

b) "Claim 6 also recites 'for each classification parameter of the rule, comparing a field of the data packet identified by a parameter ID of the classification parameter with a value of the classification parameter.' That is, there is not only a parameter ID that identifies a field of the data packet received from the applications, but there is also an associated value for comparison." (last paragraph, page 10);

c) "The Examiner cites Figure 6.7 of Stevens listing various combinations of parameters of a socket system call. However, there is no data packet from an application in a socket system call, there is no indication that there is some ID of where in the data packet the parameters of Figure 6.7 would be, and there is no indication of a value to compare against the data packet." (1st paragraph, page 11);

In response, Examiner respectfully disagrees:

a) MPEP clearly states "In certain circumstances where appropriate, an examiner may ... rely on 'common knowledge' in making a rejection" (1st paragraph, 2144.03). It is a common knowledge that "data transmitted through a stream socket has a higher priority than data transmitted through a datagram socket". For example, Hagirahim et al. (US 2002/0181401 A1) teaches "outer assigns a **priority** to the flow based on the TOS (ascertained from the packet header) and/or TCP/UDP **socket** if the flow is a non H.323 data query, the router will assign a low priority to the flow in its connection table. However, if the flow is an H.323 real time traffic flow (e.g., VoIP), the flow will be treated differently. A *real time flow*, for example, VoIP or video will be assigned to the ***highest priority*** in the connection table." ([0013]), which clearly discloses that real-time traffic which is transmitted through a stream socket has higher priority than the non real-time traffic which is transmitted through a datagram socket.

b) a socket parameters has both the parameter ID (the names of the parameters) and the value that can be used for comparison. For example, the first parameter of socket function has a parameter ID "family" and has the value of SOCK_STREAM, SOCK_DGRAM, and SOCK_RAW.

c) socket return value represents the ID of the data packets that are sent/received via the socket, as disclosed by Steven. packets associated with different sockets have different IDs.

4. For **claim 22**, Applicant argues: "The Examiner cited the IP address of a customer's house as one parameter. Office Action, p. 14. However, the Examiner fails to identify how the IP address is the only parameter. That is, if there is another parameter, such as a source address, TCP options, or the like, then there is not only one parameter." (2nd paragraph from bottom of page 10);

Applicant's argument is persuasive. Examiner has used the MAC address (instead of IP address) of the destination as the only parameter, which is consistent with the disclosure of Specification in [0065]).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. **Claims 1-4** and **24-27** are rejected under 35 U.S.C. 103(a) as being unpatentable over W. Richard Stevens, "UNIX Network Programming", 1990,

(hereinafter **Stevens**) in view of Raphaeli et al (US 20030103521, hereinafter **Raphaeli**).

For **claim 1**, Stevens discloses a method of converting application data to transport data in a communication system the method comprising:

receiving application data in a transport protocol layer from an application (data from Application layer to Transport layer, Figure 5.28, page 240) in a device (a PC, suggested by "In many PC environments", page 240) with through a service access point (a socket created by the socket System call, page 267, with description page 267-269, where the socket created by socket(int family, int *type*, int *protocol*) is an service access point), the service access point being one of a plurality of service access points of the transport protocol layer (families of socket, page 267, line 4-9 from bottom, each family provide a kind of services);

applying classifier rules associated with the service access point to the application data received through the service access point to determine if a connection through a lower protocol layer exists for the application data (socket(int *family*, ...) with *family* as classifier, page 267), including:

determining if the application data matches matching criteria of a particular classifier rule (the return code of function socket() gives indication if a connection exists (< 0) or not (>0), disclosed in C code sample in page 273, start with "if ((sockfd=socket(...))<0 ..." with the classification application data as parameters of socket functions); and

If the application data matches matching criteria of the particular classifier rule, encapsulating the application data into a payload of a transport message (application data are received from the socket identified by socket descriptor, page 269, line 5 and Figure 6.1 in page 260); and transmitting the transport messages through a connection

if the application data matches matching criteria of the particular classifier rule, encapsulating the application data into a payload of transport messages (transfer data functions: read (), write(), recv(), send(), Figure 6.1 in page 260; which encapsulate application data into transport layer message); and

transmitting the transport messages across the communication system (send(), sendto(), page 274).

Stevens **is silent on** the communication system is a power line communication system and does not explicitly disclose a higher protocol layer is serviced through a lower protocol layer.

Raphaeli teaches a power line communication system (FIG. 1, explained in [0008]) wherein a method of converting application data to transport data (application layer, [0005]) is described. Raphaeli also discloses a higher protocol layer is serviced through a lower protocol layer (FIG. 2, where a lower layer MAC provides service for upper layers). Stevens teaches IP network at network layer 3 and above, while Raphaeli discloses a specific communication system known as the power line communication system at network layer 2. One with ordinary skill in the art would have been motivated

to combine them together to provide a full network stack of the power line communication system.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to combine Stevens with Raphaeli in order to apply IP protocol to the power line communication system and providing a higher protocol layer service through a lower protocol layer.

As to **claim 2**, Stevens and Raphaeli in combination disclose the method of claim 1, Stevens further teaches the method comprising automatically establishing a connection if none exists, comprising:

generating a connection specification based upon the application data and the service access point; and establishing a connection based upon the connection specification (sample code in page 273, line 9-35, which shows to establish a connection with desired configuration parameters as the parameters socket API functions used for creating the connection) and

encapsulating the application data into transport messages for that connection (using socket API functions send(), sendto(), recv() and recvfrom(), page 274).

As to **claim 3**, Stevens and Raphaeli in combination disclose the method of claim 1, Stevens further teaches wherein receiving application data from an application further comprises receiving connection-oriented application data from the application (using socket API functions recv() and recvfrom(), page 274).

As to **claim 4**, Stevens and Raphaeli in combination disclose the method of claim 1, Stevens further teaches wherein receiving application data further comprises:

receiving connectionless application data from the application (setting up a connectionless connection via socket() with *family* parameter being set as SOCK_DGRAM, and protocol being set UDP, then using system calls recv() and recvfrom(), page 274); and encapsulating the connectionless application messages into transport data for a power line communication system connection (using socket API functions send(), sendto(), recv() and recvfrom(), page 274); wherein the power line communication system is connection-oriented (at MAC layer the power system is connection-oriented, as disclosed by Raphaeli in claim 1).

As to **claim 24**, Stevens in view of Raphaeli discloses claim 1, Stevens further discloses each classifier rule includes a priority (each rule defined in socket(int *family*, ...) in page 267 includes a corresponding priority; e.g., family value SOCK_STEAM is for real-time traffic and has a higher priority than family value SOCK_DGRAM that is for non-real-time traffic; or from to TOS field of IP header, as disclosed in Hagirahim et al. of 20020181401 A1 at [0013]); applying the rules to the data packet to the order, including when applying a particular rule to the data packet and applying the classifier rules includes applying the classifier rules to the application data in an order defined by the priorities of the classifier rules (function socket(), page 267, which implements rules to the data packet and applies the classifier rules with the priorities application data as explained above).

As to **claim 25**, Stevens in view of Raphaeli discloses claim 1, Stevens further discloses each classifier rule includes a priority (each rule defined in socket(int *family*, ...) in page 267 includes a corresponding priority; e.g., family value SOCK_STEAM is

for real-time traffic and has a higher priority than family value SOCK_DGRAM that is for non-real-time traffic); and if multiple classifier rules (rules defined in socket(int *family*, ...) in page 267) include an identical connection identifier (the return code of function socket() in page 267), applying the classifier rules includes applying the classifier rules including the identical connection identifier to the application data in an order defined by the priorities of the classifier rules including the identical connection identifier (function socket(), page 267, which implements rules to the data packet and applies the classifier rules with the priorities application data as explained above).

As to **claim 26**, Stevens in view of Raphaeli discloses claim 1, Stevens further discloses each classifier rule includes at least one classification parameter (*family* of function socket(int *family*, ...) in page 267); each classification parameter includes a parameter identification and a value (parameter *family* includes an identification *family* and a integer value); and the parameter identification identifies a field in the application data for comparison with the associated value (parameter identification *family* specifies the application data for comparison with other value, e.g., the application data associated with SOCK_STEAM are different from the application data associated with SOCK_DGRAM).

As to **claim 27**, Stevens in view of Raphaeli discloses claim 26, wherein for a classifier rule having a plurality of classification parameters (parameters of socket() in 267, such as family, type, protocol and etc.) with identical parameter identifications (the return value of socket), determining that the application data matches the classification parameters with the identical parameter identifications if the application data matches at

least one of the classification parameters with the identical parameter identifications (function socket(), page 267, which implements rules to the data packet and applies the classifier rules with the priorities application data as explained above).

7. **Claims 6-7, 9** and **21-23** are rejected under 35 U.S.C. 103(a) as being unpatentable over Andrew S. Tanenbaum, "Computer Networks", Forth edition, 8/9/2002 (hereinafter **Tanenbaum**) in view of Stevens, further in view of Kurupati et al. (US 20030182291, hereinafter Kurupati).

For **Claim 6**, Tanenbaum discloses a method of transmitting data on a network, the method comprising:

receiving an incoming data packet from an application on a device at one of a plurality of service access points of a first protocol layer (TSAP, Figure 6-8; or Lines 1-2 of first paragraph of Section 6.2.1, where a service access point of a protocol layer TSAP is considered as one of a plurality of sockets);

associating the packet with a connection (Fig. 6-8, the packet entering TSAP is associated with the connection indicated by dot line);

routing the packet to the connection ("routing packets", Lines 1-3 of first paragraph of Section 5.2) established at an interface between the first protocol layer and a second protocol layer, wherein the second protocol layer is a lower level protocol layer (Fig. 6-8, where first protocol layer is Transport layer of Host 1, and second protocol layer is Network layer of Host 2);

Tanenbaum does not explicitly discloses classifying the data packet in the first protocol layer in a classifier associated with the service access point, including:

determining an order of rules associated with the classifier to apply to the data packet using a priority of each of the rules, where each rule includes the corresponding priority; applying the rules to the data packet to the order, including when applying a particular rule to the data packet: for each classification parameter of the rule, comparing a field of the data packet identified by a parameter ID of the classification parameter with a value of the classification parameter; and if for each classification parameter of the rule, a matching value is found in the data packet, causing the packet to be associated with a connection associated with the rule.

in the same field of endeavor, Stevens discloses classifying the data packet in the first protocol layer in a classifier (parameters family, type, protocol of the socket function, page 267) associated with the service access point, including: determining an order of rules associated with the classifier to apply to the data packet using a priority of each of the rules, where each rule includes the corresponding priority (rules specified in Figure 6.7, page 268, each rule includes a corresponding priority; e.g., SOCK_STEAM normally has a higher priority than SOCK_DGRAM); applying the rules to the data packet to the order, including when applying a particular rule to the data packet (function socket(), page 267, which implements rules according to parameters of the function): for each classification parameter of the rule, comparing a field of the data packet identified by a parameter ID of the classification parameter with a value of the classification parameter (note that socket return value represents the ID of the data packets that are sent/received via the socket, as disclosed by Steven. packets associated with different sockets have different IDs, see C code sample in page 273);

and if for each classification parameter of the rule, a matching value is found in the data packet, causing the packet to be associated with a connection associated with the rule (parameters and associated rules specified in Figure 6.7, page 268 for function socket to implement on the packet with associated connection).

Stevens simply teaches details of the socket that is disclosed by Tanenbaum, therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to combine modify socket by Tanenbaum with the detailed socket features disclosed by Stevens to provide desired network service.

Tanenbaum in view of Stevens does not explicitly disclose using priority of each of the rules.

Kurupati discloses each of the rules is associated with priority ("A rule is typically associated with each IP address ... The rule associated with an IP address indicates what action--e.g., a routing decision, a network address translation, a **priority** determination, and/or a filtering function--is to be taken with respect to a packet having that IP address", [0017]).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to combined Tanenbaum in view of Stevens with Kurupati to use the rules associated with priority to more ensure data being delivered following priority.

As to **claim 7**, Tanenbaum and Stevens and Kurupati in combination disclose the method of claim 6, Tanenbaum further teaches the method comprising fragmenting the

packet into smaller packets as needed based upon the packet size ("maximum packet size, … break it into 4 packets", Section 5.1.3).

As to **claim 9**, Tanenbaum and Stevens and Kurupati in combination disclose the method of claim 6, Tanenbaum teaches classifying the data packet further comprising determining if a connection exists for the packet, and requesting a connection if a connection does not exist (Section 6.1.4, the server code section of /* Passive open, Wait for connection. */, wherein [s=socket(…), if (s<0) fatal(socket failed) …] suggest that if the value of s < 0, the socket exists already; otherwise, it successfully creates a socket for a new connection).

As to **claim 21**, Tanenbaum in view of Stevens and Kurupati discloses the method of claim 6, Tanenbaum further discloses the method comprising; a connection identifier (socket parameter *type*, Section 6.1.3, which is a part of connection identifier of socket, has the value of SOCK_STREAM, SOCK_DGRAM, SOCK_RAW and etc, with SOCK_STREAM has the highest priority, page 268, line 7-13, Steven); a transport layer port (each transport layer port represents an application, for example, a TCP port 23 for telnet has a higher priority than a TCP port 25 for SMTP used by e-mail, Section 8.6.2 and 7.4.4 of Tanenbaum); and at least one classification parameter, each classification parameter including a parameter ID and a value (IP destination address and IP destination Port; Section 6.5.4; this is the same as disclosed in the specification in [0062]).

As to **claim 22**, Tanenbaum in view of Stevens and Kurupati discloses the method of claim 21, Tanenbaum further discloses each rule associated with audio/visual

application data (suggested by "how computer process audio and video", Section 7.4, last paragraph), the rule includes only one classification parameter (the MAC address of the device for "a video on demand system" shown in Figure 7-78, Section 7.4.8; Note that MAC/Ethernet address is unique for each device (3rd paragraph of Section 4.3.3) and is consistent with the application specification [0065]).

As to **claim 23**, Tanenbaum in view of Stevens and Kurupati discloses the method of claim 22, Tanenbaum further discloses each rule associated with audio/visual application data (Figure 7-78, Video-on-demand system), the classification parameter of the rule includes a destination address as the parameter ID (the IP address of a customer's house for "a video on demand system" shown in Figure 7-78, Section 7.4.8).

8. **Claims 8** is rejected under 35 U.S.C. 103(a) as being unpatentable over S. Tanenbaum in view Stevens and Kurupati, further in view of Malkin (US 6272145 B1).

As to **claim 8**, Tanenbaum in view Stevens and Kurupati discloses the method of claim 6, the method comprising fragmenting the packet into smaller packets as needed ("maximum packet size, ... break it into 4 packets", Section 5.1.3).

Tanenbaum **does not** explicitly teach that the fragmenting depends upon the bandwidth of the connection.

In the same field of endeavor, Malkin discloses the fragmenting depends upon the bandwidth of the connection ("size of the different fragment will vary depending on ... bandwidth of each link", col. 7, line 26-28).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to fragmenting depends upon the bandwidth of the connection for the benefit of efficiency and quality of service enhancement of network operation.

9. **Claims 11** and **13** are rejected under 35 U.S.C. 103(a) as being unpatentable over Stevens in view of Hogan et al. (US Patent 4,841,456, hereinafter Hogan).

For **Claim 11**, Stevens discloses a method of classifying data packets in a communication system, the method comprising:

analyzing an incoming data packet according to a plurality of sets of parameters (sets of socket parameters, such as *family, type, protocol*, and etc, Page 267, different types of socket has different types of parameters), wherein the sets of parameters analyzed depends upon a type of service access point (socket *type* of socket system call, Page 267) from which the data packet came, and the sets of parameters are used in analyzing the data packet according to an order of the priorities of the sets of parameters (parameters of socket functions, such as *family*, *type* and *protocol,* Page 267-268); if the set of parameters in the data packet match a predefined set of parameters associated with connection, associating a connection (a connection is identified by socket descriptor, page 269, line 5) for the predefined set of parameters with the packet (the parameters of socket functions).

Steven does not explicitly disclose each set of parameters includes a priority;

Hogan discloses apply priority of to each of the rules ("A typical priority rule might be that the rule having the highest number of conditions (i.e., a specific rule) has priority

over a rule having a smaller number of conditions", col. 8, line 28-31). The Hogan's teaching is very general and applies to any set of rules.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to combine Stevens with Hogan in order to apply socket API to Internet ([0016]).

As to **claim 13**, Stevens in view of Hogan discloses the method of claim 11, Stevens further discloses that the method comprising transmitting parameters of the data packet to a connection manager if the parameters of the data packet do not match a predefined set of parameters (page 283, line 5-6, if (sendto(sockfd, mesg, n, 0, pcli_addr, clilen) !=n) err_dump("dg_echo: sendto error"); where the connection manager is the Operating System, the n bytes of data to be sent to specified socketfd and pcli_addr must match with the number of data byte sent).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jianye Wu whose telephone number is (571)270-1665. The examiner can normally be reached on Monday to Thursday, 8am to 7pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Seema Rao can be reached on (571)272-3174. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jianye Wu/

Examiner, Art Unit 2462